

APPARATUS AND METHOD FOR TRACE
STREAM IDENTIFICATION OF MULTIPLE
TARGET PROCESSOR EVENTS

This application claims priority under 35 USC §119 (e) (1) of Provisional Application Number 60/434,174 (TI-34671P) filed December 17, 2002.

Related Applications

- 5 U.S. Patent Application (Attorney Docket No. TI-34654),
entitled APPARATUS AND METHOD FOR SYNCHRONIZATION OF TRACE
STREAMS FROM MULTIPLE PROCESSORS, invented by Gary L.
Swoboda, filed on even date herewith, and assigned to the
assignee of the present application; U.S. Patent
10 Application (Attorney Docket No. TI-34655), entitled
APPARATUS AND METHOD FOR SEPARATING DETECTION AND ASSERTION
OF A TRIGGER EVENT, invented by Gary L. Swoboda, filed on

5 even date herewith, and assigned to the assignee of the present application; U.S. Patent Application (Attorney Docket No. TI- 34656), entitled APPARATUS AND METHOD FOR STATE SELECTABLE TRACE STREAM GENERATION, invented by Gary L. Swoboda, filed on even date herewith, and assigned to
10 the assignee of the present application; U.S. Patent Application (Attorney Docket No. TI-34657), entitled APPARATUS AND METHOD FOR SELECTING PROGRAM HALTS IN AN UNPROTECTED PIPELINE AT NON-INTERRUPTIBLE POINTS IN CODE EXECUTION, invented by Gary L. Swoboda and Krishna Allam,
15 filed on even date herewith, and assigned to the assignee of the present application; U.S. Patent Application (Attorney Docket No. TI-34658), entitled APPARATUS AND METHOD FOR REPORTING PROGRAM HALTS IN AN UNPROTECTED PIPELINE AT NON-INTERRUPTIBLE POINTS IN CODE EXECUTION,
20 invented by Gary L. Swoboda, filed on even date herewith, and assigned to the assignee of the present application; U.S. Patent Application (Attorney Docket No. TI-34659), entitled APPARATUS AND METHOD FOR A FLUSH PROCEDURE IN AN INTERRUPTED TRACE STREAM, invented by Gary L. Swoboda,
25 filed on even date herewith, and assigned to the assignee of the present application; U.S. Patent Application (Attorney Docket No. TI-34660), entitled APPARATUS AND METHOD FOR CAPTURING AN EVENT OR COMBINATION OF EVENTS RESULTING IN A TRIGGER SIGNAL IN A TARGET PROCESSOR,
30 invented by Gary L. Swoboda, filed on even date herewith, and assigned to the assignee of the present application;

5 U.S. Patent Application (Attorney Docket No. TI-34661),
entitled APPARATUS AND METHOD FOR CAPTURING THE PROGRAM
COUNTER ADDRESS ASSOCIATED WITH A TRIGGER SIGNAL IN A
TARGET PROCESSOR, invented by Gary L. Swoboda, filed on
even date herewith, and assigned to the assignee of the
10 present application; U.S. Patent Application (Attorney
Docket No. TI-34662), entitled APPARATUS AND METHOD
DETECTING ADDRESS CHARACTERISTICS FOR USE WITH A TRIGGER
GENERATION UNIT IN A TARGET PROCESSOR, invented by Gary
Swoboda and Jason L. Peck, filed on even date herewith, and
15 assigned to the assignee of the present application; U.S.
Patent Application (Attorney Docket No. TI-34663), entitled
APPARATUS AND METHOD FOR TRACE STREAM IDENTIFICATION OF A
PROCESSOR RESET, invented by Gary L. Swoboda, Bryan Thome
and Manisha Agarwala, filed on even date herewith, and
20 assigned to the assignee of the present application; U.S.
Patent (Attorney Docket No. TI-34664), entitled APPARATUS
AND METHOD FOR TRACE STREAM IDENTIFICATION OF A PROCESSOR
DEBUG HALT SIGNAL, invented by Gary L. Swoboda, Bryan
Thome, Lewis Nardini and Manisha Agarwala, filed on even
25 date herewith, and assigned to the assignee of the present
application; U.S. Patent Application (Attorney Docket No.
TI-34665), entitled APPARATUS AND METHOD FOR TRACE STREAM
IDENTIFICATION OF A PIPELINE FLATTENER PRIMARY CODE FLUSH
FOLLOWING INITIATION OF AN INTERRUPT SERVICE ROUTINE;
30 invented by Gary L. Swoboda, Bryan Thome and Manisha
Agarwala, filed on even date herewith, and assigned to the

5 assignee of the present application; U.S. Patent
Application (Attorney Docket No. TI-34666), entitled
APPARATUS AND METHOD FOR TRACE STREAM IDENTIFICATION OF A
PIPELINE FLATTENER SECONDARY CODE FLUSH FOLLOWING A RETURN
TO PRIMARY CODE EXECUTION, invented by Gary L. Swoboda,
10 Bryan Thome and Manisha Agarwala filed on even date
herewith, and assigned to the assignee of the present
application; U.S. Patent Application (Docket No. TI-34667),
entitled APPARATUS AND METHOD IDENTIFICATION OF A PRIMARY
CODE START SYNC POINT FOLLOWING A RETURN TO PRIMARY CODE
15 EXECUTION, invented by Gary L. Swoboda, Bryan Thome and
Manisha Agarwala, filed on even date herewith, and assigned
to the assignee of the present application; U. S. Patent
Application (Attorney Docket No. TI-34668), entitled
APPARATUS AND METHOD FOR IDENTIFICATION OF A NEW SECONDARY
20 CODE START POINT FOLLOWING A RETURN FROM A SECONDARY CODE
EXECUTION, invented by Gary L. Swoboda, Bryan Thome and
Manisha Agarwala, filed on even date herewith, and assigned
to the assignee of the present application; U.S. Patent
Application (Attorney Docket No. TI-34669), entitled
25 APPARATUS AND METHOD FOR TRACE STREAM IDENTIFICATION OF A
PAUSE POINT IN A CODE EXECUTION SEQUENCE, invented by Gary
L. Swoboda, Bryan Thome and Manisha Agarwala, filed on even
date herewith, and assigned to the assignee of the present
application; U.S. Patent Application (Attorney Docket No.
30 TI-34670), entitled APPARATUS AND METHOD FOR COMPRESSION OF
A TIMING TRACE STREAM, invented by Gary L. Swoboda and

5 Bryan Thome, filed on even date herewith, and assigned to
the assignee of the present application; and U.S. Patent
Application (Attorney Docket No. TI-34672 entitled
APPARATUS AND METHOD FOR OP CODE EXTENSION IN PACKET GROUPS
TRANSMITTED IN TRACE STREAMS, invented by Gary L. Swoboda
10 and Bryan Thome, filed on even date herewith, and assigned
to the assignee of the present application are related
applications.

15 **Background of the Invention**

1. Field of the Invention

This invention relates generally to the testing of digital
20 signal processing units and, more particularly, to the
signals that are transmitted from a target processor to a
host processing to permit analysis of the target processing
unit operation. Certain events in the target processor
must be communicated to the host processing unit along with
25 contextual information. In this manner, the test and debug
data can be analyzed and problems in the operation of the
target processor identified.

2. Description of the Related Art

30

As microprocessors and digital signal processors have
become increasingly complex, advanced techniques have been

5 developed to test these devices. Dedicated apparatus is available to implement the advanced techniques. Referring to Fig. 1A, a general configuration for the test and debug of a target processor **12** is shown. The test and debug procedures operate under control of a host processing unit

10 **10**. The host processing unit **10** applies control signals to the emulation unit **11** and receives (test) data signals from the emulation unit **11** by cable connector **14**. The emulation unit **11** applies control signals to and receives (test) signals from the target processing unit **12** by connector

15 cable **15**. The emulation unit **11** can be thought of as an interface unit between the host processing unit **10** and the target processor **12**. The emulation unit **11** processes the control signals from the host processor unit **10** and applies these signals to the target processor **12** in such a manner

20 that the target processor will respond with the appropriate test signals. The test signals from the target processor **12** can be a variety types. Two of the most popular test signal types are the JTAG (Joint Test Action Group) signals and trace signals. The JTAG protocol provides a

25 standardized test procedure in wide use in which the status of selected components is determined in response to control signals from the host processing unit. Trace signals are signals from a multiplicity of selected locations in the target processor **12** during defined period of operation.

30 While the width of the bus **15** interfacing to the host processing unit **10** generally has a standardized dimension,

5 the bus between the emulation unit **11** and the target processor **12** can be increased to accommodate an increasing amount of data needed to verify the operation of the target processing unit **12**. Part of the interface function between the host processing unit **10** and the target processor **12** is
10 to store the test signals until the signals can be transmitted to the host processing unit **10**.

Referring to Fig. 1B, the operation of the trigger generation unit **19** is shown. The trigger unit provides the
15 main component by which the operation/state of the target processor can be altered. At least one event signal is applied to the trigger generation unit **19**. Based on the identity of the event signal(s) applied to the trigger generation unit **19**, a trigger signal is selected. Certain
20 events and combination of events, referred to as an event front, generate a selected trigger signal that results in certain activity in the target processor, e.g., as a debug halt. Combinations of different events generating trigger signals are referred to as jobs. Multiple jobs can have
25 the same trigger signal or combination of trigger signals. In the test and debug of the target processor, the trigger signals can provide impetus for changing state in the target processor or for performing a specified activity. The event front defines the reason for the generation of
30 trigger signal. This information is important in understanding the operation of the target processor

5 because, as pointed out above, several combinations of
events can result in the generation of the same trigger
signal. In order to analyze the operation of the target
processing unit, the portion of the code resulting in the
trigger signal must be identified. However, the events in
10 the host processor leading to the generation of event
signals can be complicated. Specifically, the
characteristics of an instruction at a program counter
address can determine whether a trigger signal should be
generated. A trigger signal can be an indication of when
15 an address is within a range of addresses, outside of a
range of addresses, some combination of address
characteristics, and/or the address is aligned with a
reference address. In this instance, the address can be
the program address of an instruction or a memory address
20 directly or indirectly referenced by a program instruction.

As will be seen, event signals can come from several
sources. And these event signals provide a plurality of
related signals that should be included in a trace stream
25 in order to reconstruct the target processor activity. In
addition, event signals and other signals that must be
communicated to the host processing unit may occur
simultaneously. The communication of the simultaneous
occurrence of conditions in the target processor should be
30 performed with a minimum of additional equipment and with a
minimum impact on the band width of the trace streams.

5

A need has been felt for apparatus and an associated method having the feature that a plurality of simultaneous target processor events can be communicated to the host processing unit. It is another feature of the apparatus and
10 associated method to communicate occurrence of the simultaneous events to the host processing unit using trace stream sync marker procedures. It is a still further feature of the apparatus and associated method to communicate to the host processing unit the occurrence of a
15 plurality of simultaneous events in the target processor using a minimum of information. It is yet another object of the apparatus and associated method to communicate the occurrence of a plurality of simultaneous events with a minimum of new apparatus.

20

5 **Summary of the Invention**

The aforementioned and other features are accomplished, according to the present invention, by providing the target processor with at least two trace streams. One of the
10 trace streams is a timing trace stream. The second trace stream, when an event is identified, is provided with a sync marker. The sync marker includes at least one portion identifying the event resulting in the sync marker, a portion relating the event to the timing trace stream, and
15 a portion identifying the point in the program execution when the event is identified. The trace streams are implemented with signal packets. When more than one simultaneous event is to be communicated to the host processing unit, an additional packet in the storage unit
20 in which the sync marker is assembled is activated. Each of the locations in the additional packet can be associated with one event. During the assembly of the sync marker, one event is identified in the normal fashion, i.e., the header of a packet sync marker group. The remaining event
25 or events have a logic signal stored in the associated location in the additional packet. After the assembly, the expanded sync marker is transferred to the host processing unit. In the preferred embodiment, the second trace stream is a program counter trace stream. The point in the
30 program execution where the events are identified is determined by the program counter address included in the

- 5 sync marker. The time of the occurrence of the events in the target processor is determined by trace synchronization markers and by a position of a clock cycle in a timing packet.
- 10 Other features and advantages of present invention will be more clearly understood upon reading of the following description and the accompanying drawings and the claims.

Brief Description of the Drawings

5

Figure 1A is a general block diagram of a system configuration for test and debug of a target processor, while Fig. 1B illustrates a trigger unit in the target processor.

10

Figure 2 is a block diagram of selected components in the target processor used the testing of the central processing unit of the target processor according to the present invention.

15

Figure 3 is a block diagram of selected components of the illustrating the relationship between the components transmitting trace streams in the target processor.

20

Figure 4A illustrates format by which the timing packets are assembled according to the present invention, while Figure 4B illustrates the inclusion of a periodic sync marker in the timing trace stream.

25

Figure 5 illustrates the parameters for sync markers in the program counter stream packets according to the present invention.

Figure 6A illustrates the sync markers in the program counter trace stream when a periodic sync ID signal is

30

5 generated, while Figure 6B illustrates the reconstruction of the target processor operation from the trace streams according to the present invention.

Figure 7 is a block diagram illustrating the apparatus used
10 in reconstructing the processor operation from the trace streams according to the present invention.

Figure 8A is block diagram of the program counter sync marker generator unit; Figure 8B illustrates the additional
15 apparatus needed to form a sync marker for a plurality of events, Figure 8C illustrates the multiple-event sync marker in the trace stream, and Figure 8D illustrates the reconstruction of the trace streams by the host processing unit.

20

Description of the Preferred Embodiment

1. Detailed Description of the Figures

25 Fig. 1A and Fig. 1B have been described with respect to the related art.

Referring to Fig. 2, a block diagram of selected components of a target processor **20**, according to the present
30 invention, is shown. The target processor includes at least one central processing unit **200** and a memory unit

5 **208.** The central processing unit **200** and the memory unit
 208 are the components being tested. The trace system for
 testing the central processing unit **200** and the memory unit
 202 includes three packet generating units, a data packet
 generation unit **201**, a program counter packet generation
10 unit **202** and a timing packet generation unit **203**. The data
 packet generation unit **201** receives VALID signals,
 READ/WRITE signals and DATA signals from the central
 processing unit **200**. After placing the signals in packets,
 the packets are applied to the scheduler/multiplexer unit
15 **204** and forwarded to the test and debug port **205** for
 transfer to the emulation unit **11**. The program counter
 packet generation unit **202** receives PROGRAM COUNTER
 signals, VALID signals, BRANCH signals, and BRANCH TYPE
 signals from the central processing unit **200** and, after
20 forming these signal into packets, applies the resulting
 program counter packets to the scheduler/multiplexer **204**
 for transfer to the test and debug port **205**. The timing
 packet generation unit **203** receives ADVANCE signals, VALID
 signals and CLOCK signals from the central processing unit
25 **200** and, after forming these signal into packets, applies
 the resulting packets to the scheduler/multiplexer unit **204**
 and the scheduler/multiplexer **204** applies the packets to
 the test and debug port **205**. Trigger unit **209** receives
 EVENT signals from the central processing unit **200** and
30 signals that are applied to the data trace generation unit
 201, the program counter trace generation unit **202**, and the

5 timing trace generation unit **203**. The trigger unit **209**
applies TRIGGER and CONTROL signals to the central
processing unit **200** and applies CONTROL (i.e., STOP and
START) signals to the data trace generation unit **201**, the
program counter generation unit **202**, and the timing trace
10 generation unit **203**. The sync ID generation unit **207**
applies signals to the data trace generation unit **201**, the
program counter trace generation unit **202** and the timing
trace generation unit **203**. While the test and debug
apparatus components are shown as being separate from the
15 central processing unit **201**, it will be clear that an
implementation these components can be integrated with the
components of the central processing unit **201**.

Referring to Fig. 3, the relationship between selected
20 components in the target processor **20** is illustrated. The
data trace generation unit **201** includes a packet assembly
unit **2011** and a FIFO (first in/first out) storage unit
2012, the program counter trace generation unit **202**
includes a packet assembly unit **2021** and a FIFO storage
25 unit **2022**, and the timing trace generation unit **203**
includes a packet generation unit **2031** and a FIFO storage
unit **2032**. As the signals are applied to the packet
generators **201**, **202**, and **203**, the signals are assembled
into packets of information. The packets in the preferred
30 embodiment are 10 bits in width. Packets are assembled in
the packet assembly units in response to input signals and

5 transferred to the associated FIFO unit. The scheduler/multiplexer **204** generates a signal to a selected trace generation unit and the contents of the associated FIFO storage unit are transferred to the scheduler/multiplexer **204** for transfer to the emulation
10 unit. Also illustrated in Fig. 3 is the sync ID generation unit **207**. The sync ID generation unit **207** applies an SYNC ID signal to the packet assembly unit of each trace generation unit. The periodic signal, a counter signal in the preferred embodiment, is included in a current packet
15 and transferred to the associated FIFO unit. The packet resulting from the SYNC ID signal in each trace is transferred to the emulation unit and then to the host processing unit. In the host processing unit, the same sync marker ID in each trace stream indicates that the
20 point at which the trace streams are synchronized. In addition, the packet assembly unit **2031** of the timing trace generation unit **203** applies an INDEX signal to the packet assembly unit **2021** of the program counter trace generation unit **202**. The function of the INDEX signal will be
25 described below.

Referring to Fig. 4A, the assembly of timing packets is illustrated. The signals applied to the timing trace generation unit **203** are the CLOCK signals and the ADVANCE
30 signals. The CLOCK signals are system clock signals to which the operation of the central processing unit **200** is

5 synchronized. The ADVANCE signals indicate an activity
such as a pipeline advance or program counter advance (())
or a pipeline non-advance or program counter non-advance
(1). An ADVANCE or NON-ADVANCE signal occurs each clock
cycle. The timing packet is assembled so that the logic
10 signal indicating ADVANCE or NON-ADVANCE is transmitted at
the position of the concurrent CLOCK signal. These
combined CLOCK/ADVANCE signals are divided into groups of 8
signals, assembled with two control bits in the packet
assembly unit **2031**, and transferred to the FIFO storage
15 unit **2032**.

Referring to Fig. 4B, the trace stream generated by the
timing trace generation unit **203** is illustrated. The first
(in time) trace packet is generated as before. During the
20 assembly of the second trace packet, a SYNC ID signal is
generated during the third clock cycle. In response, the
timing packet assembly unit **2031** assembles a packet in
response to the SYNC ID signal that includes the sync ID
number. The next timing packet is only partially assembled
25 at the time of the SYNC ID signal. In fact, the SYNC ID
signal occurs during the third clock cycle of the formation
of this timing packet. The timing packet assembly unit
2031 generates a TIMING INDEX 3 signal (for the third
packet clock cycle at which the SYNC ID signal occurs) and
30 transmits this TIMING INDEX 3 signal to the program counter
packet assembly unit **2031**.

5 Referring to Fig. 5, the parameters of a sync marker in the
program counter trace stream, according to the present
invention is shown. The program counter stream sync
markers each have a plurality of packets associated
therewith. The packets of each sync marker can transmit a
10 plurality of parameters. A SYNC POINT TYPE parameter
defines the event described by the contents of the
accompanying packets. A program counter TYPE FAMILY
parameter provides a context for the SYNC POINT TYPE
parameter and is described by the first two most
15 significant bits of a second header packet. A BRANCH INDEX
parameter in all but the final SYNC POINT points to a bit
within the next relative branch packet following the SYNC
POINT. When the program counter trace stream is disabled,
this index points a bit in the previous relative branch
20 packet when the BRANCH INDEX parameter is not a logic "0".
In this situation, the branch register will not be complete
and will be considered as flushed. When the BRANCH INDEX
is a logic "0", this value point to the least significant
value of branch register and is the oldest branch in the
25 packet. A SYNC ID parameter matches the SYNC POINT with
the corresponding TIMING and/or DATA SYNC POINT which are
tagged with the same SYNC ID parameter. A TIMING INDEX
parameter is applied relative to a corresponding TIMING
SYNC POINT. For all but LAST POINT SYNC events, the first
30 timing packet after the TIMING PACKET contains timing bits
during which the SYNC POINT occurred. When the timing

5 stream is disabled, the TIMING INDEX points to a bit in the timing packet just previous to the TIMING SYNC POINT packet when the TIMING INDEX value is not zero. In this situation, the timing packet is considered as flushed. A TYPE DATA parameter is defined by each SYNC TYPE. An
10 ABSOLUTE PC VALUE is the program counter address at which the program counter trace stream and the timing information are aligned. An OFFSET COUNT parameter is the program counter offset counter at which the program counter and the timing information are aligned.

15

Referring to Fig. 6A, a program counter trace stream for a hypothetical program execution is illustrated. In this program example, the execution proceeds without interruption from external events. The program counter
20 trace stream will consist of a first sync point marker 601, a plurality of periodic sync point ID markers 602, and last sync point marker 603 designating the end of the test procedure. The principal parameters of each of the packets are a sync point type, a sync point ID, a timing index, and
25 an absolute PC value. The first and last sync points identify the beginning and the end of the trace stream. The sync ID parameter is the value from the value from the most recent sync point ID generator unit. In the preferred embodiment, this value is a 3-bit logic sequence. The
30 timing index identifies the status of the clock signals in a packet, i.e., the position in the 8 position timing

5 packet when the event producing the sync signal occurs. And the absolute address of the program counter at the time that the event causing the sync packet is provided. Based on this information, the events in the target processor can be reconstructed by the host processor.

10

Referring to Fig. 6B, the reconstruction of the program execution from the timing and program counter trace streams is illustrated. The timing trace stream consists of packets of 8 logic "0"s and logic "1"s. The logic "0"s
15 indicate that either the program counter or the pipeline is advanced, while the logic "1"s indicate the either the program counter or the pipeline is stalled during that clock cycle. Because each program counter trace packet has an absolute address parameter, a sync ID, and the timing
20 index in addition to the packet identifying parameter, the program counter addresses can be identified with a particular clock cycle. Similarly, the periodic sync points can be specifically identified with a clock cycle in the timing trace stream. In this illustration, the timing
25 trace stream and the sync ID generating unit are in operation when the program counter trace stream is initiated. The periodic sync point is illustrative of the plurality of periodic sync points that would typically be available between the first and the last trace point, the
30 periodic sync points permitting the synchronization of the three trace streams for a processing unit.

5

Referring to Fig. 7A, the general technique for reconstruction of the trace streams is illustrated. The trace streams originate in the target processor **12** as the target processor **12** is executing a program **1201**. The trace signals are applied to the host processing unit **10**. The host processing unit **10** also includes the same program **1201**. Therefore, in the illustrative example of Fig. 6 wherein the program execution proceeds without interruptions or changes, only the first and the final absolute addresses of the program counter are needed. Using the advance/non-advance signals of the timing trace stream, the host processing unit can reconstruct the program as a function of clock cycle. Therefore, without the sync ID packets, only the first and last sync markers are needed for the trace stream. This technique results in reduced information transfer. Fig. 6 includes the presence of periodic sync ID cycles, of which only one is shown. The periodic sync ID packets are important for synchronizing the plurality of trace streams, for selection of a particular portion of the program to analyze, and for restarting a program execution analysis for a situation wherein at least a portion of the data in the trace data stream is lost. The host processor can discard the (incomplete) trace data information between two sync ID packets and proceed with the analysis of the program outside of the sync timing packets defining the lost data.

30

5

As indicated in Fig. 6A, the program counter trace stream includes the absolute address of the program counter for an instruction. Referring to Fig. 7B, each processor includes a processor pipeline **71**. When the instruction leaves the processor pipeline, the instruction is entered in the pipeline flattener **73**. At the same time, an access of memory unit **72** is performed. The results of the memory access of memory unit **72**, which may take several clock cycles, is then merged the associated instruction in the pipeline flattener **73** and withdrawn from the pipeline flattener **73** for appropriate distribution. The pipeline flattener **73** provides a technique for maintaining the order of instructions while providing for the delay of a memory access. In the preferred embodiment, the absolute address used in the program counter trace stream is the derived from the instruction of leaving the pipeline flattener **71**. As a practical matter, the absolute address is delayed. It is not necessary to include a pipeline flattener **73** in the target processor. The instructions can have appropriate labels associated therewith to eliminate the need for the pipeline flattener **73**.

Referring to Fig. 8A, the major components of the program counter packet generation unit **202** illustrating the generation of the program counter trace stream is shown. The program counter packet generation unit **202** includes a

5 decoder unit **2023**, storage unit **2021**, a FIFO unit **2022**, and
a gate unit **2024**. PERIODIC SYNC ID signals, TIMING INDEX
signals, and ABSOLUTE ADDRESS signals are applied to gate
unit **2024**. When the PERIODIC SYNC ID signals are
incremented, the decoder unit **2023**, in response to the
10 PERIODIC SYNC ID signal, stores a periodic sync ID header
signal group in a predetermined location **2021A** of the
header portion of the storage unit **2021**. The PERIODIC SYNC
signal causes the gate **2024** to transmit the PERIODIC SYNC
ID signals, the TIMING INDEX signals and the ABSOLUTE
15 ADDRESS signals. These transmitted signals are stored in
the storage unit **2021** in information packet locations
assigned to these parameters. When all of the portions of
the periodic sync marker have been assembled in the storage
unit **2021**, then the component packets of the periodic sync
20 marker are transferred to the FIFO unit **2022** for eventual
transmission to the scheduler/multiplexer unit. Similarly,
when another event signal is generated and applied to the
decoder unit **2023**, the reset header identifying the event
signal is stored in position **2021A** in the header portion of
25 the storage unit **2021**. The event signal applied to decoder
unit **2023** results in a control signal being applied to the
gate **2024**. As a result of the control signal, the SYNC ID
signals, the TIMING INDEX signals, and the ABSOLUTE ADDRESS
signals are stored in the appropriate locations in storage
30 unit **2021**. When the event signal sync marker has been

5 assembled, i.e., in packets, the sync marker is transferred to the FIFO unit **2022**.

Referring to fig. 8B, the additional apparatus needed to accommodate a multiple-event sync signal is shown. As
10 before, one event is identified by related signals stored in the **2021A** locations of the header packets (i.e., storage locations). Multiple-event gate **2027** has applied thereto any remaining event signal. When more than one event signal is applied to the decoder unit **2023**, a control
15 signal is applied to the multiple-event gate **2027** and the event signals are applied to locations in the storage locations (packet) **2012B**. Each event signal has a specified location in packet **2012B**. When an event signal is active, a logic signal is stored in the associated
20 location in packet **2021B**. In addition, in response to the control signal, the multiple-event gate applies an op code to locations in the packet **2021B**. This op code identifies the packet as being a continuation of the sync marker being assembled in the storage unit **2021**. When the sync marker
25 is transferred to the FIFO unit **2022**, the additional packet **2021B** is transmitted therewith.

Referring to Fig. 8C, examples of the sync markers in the program counter trace stream are shown. The start of the
30 test procedure is shown in first point sync marker 801. Thereafter, periodic sync ID markers 805 can be generated.

5 Other event markers can also be generated. The identification of multiple simultaneous events results in the generation of multiple-event sync marker 810.

Referring to Fig. 8D, the reconstruction of the program
10 counter trace stream from the sync markers is shown. The first sync point marker indicates the beginning of test procedure with a program counter address of PC. The program continues to execute unit with the program counter addresses being related to a particular processor clock
15 cycle. In the clock cycle following the program counter address PC+3, a periodic sync ID signal is generated. At program counter address PC+5, the multiple-event sync marker is generated.

20 2. Operation of the Preferred Embodiment

The present invention relies on the ability to of the host processing unit to relate the timing trace stream and the program counter trace stream. This relationship is
25 provided by having periodic sync ID information transmitted in each trace stream. In addition, the timing packets are grouped in packets of eight signals identifying whether the program counter or the pipeline advanced or did not advance. The sync markers in the program counter stream
30 include both the periodic sync ID information and an index indicating the position in the current eight-position

5 timing packet when the event occurred. Thus, the clock
cycle of the event can be specified. In addition, the
address of the program counter is provided in the program
counter sync markers so that the event can be related to
the execution of the program. As a result, when a
10 multiple-event sync marker is generated, the location of
the events relative to the target processor clock and to
the program execution is established and program execution
of the target processor can be reconstructed. In the
preferred embodiment, data (memory access) information is
15 also transferred from the target processor to the host
processing unit in a trace data stream. It is therefore
possible to reconstruct the entire operation of the target
processor from the transmitted trace streams.

20 The sync marker trace streams illustrated above relate to an
idealized operation of the target processor in order to
emphasize the features of the present invention. Numerous
other sync events (e.g. branch events) will typically be
entered in the program counter trace stream generation unit
25 and included in the program counter trace stream.

In the foregoing discussion, the sync markers can have
additional information embedded therein depending on the
implementation of the apparatus generating and interpreting
30 the trace streams. This information will be related to the
parameters shown in Fig. 5. It will also be clear that a

5 data trace stream, as shown in Fig. 2 will typically be present. The periodic sync IDs as well as the timing indexes will also be included in the data trace stream. In addition, the program counter absolute address parameter can be replaced by the program counter off-set register in
10 certain situations.

While the invention has been described with respect to the embodiments set forth above, the invention is not necessarily limited to these embodiments. Accordingly,
15 other embodiments, variations, and improvements not described herein are not necessarily excluded from the scope of the invention, the scope of the invention being defined by the following claims.